Claims

What is claimed is:

[c1] A network system for key management, comprising:

a server;

a key management system providing process logic for key management system management located on the server;

a key management system storage providing a secure data storage for the key management system;

an application using the key management system to manage an application key; and

an interface providing a means for managing the key management system.

[c2] The network system of claim 1, further comprising:

a client computer operatively connected to the server, wherein the client computer comprises a user interface to manage the key management system.

[c3] The network system of claim 1, wherein the key management storage is located on the server.

[c4] The network system of claim 1, wherein the key management storage is located on a second server operatively connected to the server.

[c5] The network system of claim 1, wherein the interface comprises a graphical user interface.

[c6] The network system of claim 5, wherein the graphical user interface is integrated into a web browser.

[c7] The network system of claim 2, wherein the user interface comprises a graphical user interface.

[c8] The network system of claim 7, wherein the graphical user interface is integrated into a web browser.

[c9] The network system of claim 2, wherein the client computer and the server are connected using an encrypted connection.

[c10] The network system of claim 1, wherein the key management system further comprises:

a memory storing data within the key management system;

a hashing module hashing a key encryption key;

an encryption module decrypting data; and

a serialization module de-serializing data obtained from the memory, the encryption module, and the serialization module.

[c11] The network system of claim 1, wherein the key management system further comprises:

a memory storing data within the key management system;

a hashing module hashing a key encryption key;

an encryption module decrypting data and encrypting data; and

a serialization module de-serializing and serializing data obtained from the memory, the encryption module, and the serialization module.

[c12] The key management system of claim 10, further comprising:

an encoding module for encoding data.

[c13] The key management system of claim 10, wherein the hashing module uses an MD5 hashing function.

[c14] The key management system of claim 10, wherein the encryption module further comprises a key generation tool.

[c15] The key management system of claim 14, wherein the key generation tool comprises a symmetric algorithm.

[c16] The key management system of claim 14, wherein the key generation tool comprises an asymmetric algorithm.

[c17] The key management system of claim 11, further comprising:
an encoding module for encoding data.

[c18] The key management system of claim 11, wherein the hashing module uses an MD5 hashing function.

[c19] The key management system of claim 11, wherein the encryption module further comprises a key generation tool.

[c20] The key management system of claim 19, wherein the key generation tool comprises a symmetric algorithm.

[c21] The key management system of claim 19, wherein the key generation tool comprises an asymmetric algorithm.

[c22] The key management system of claim 1, wherein the interface comprises a means for changing a key encryption key.

[c23] The key management system of claim 1, wherein the interface comprises means for starting the key management system.

[c24] The key management system of claim 1, wherein the interface comprises means for initializing the key management system.

[c25] The key management system of claim 1, wherein the interface comprises means for diagnosing problems with the key management system.

[c26] A network system for key management, comprising:

a server;

a key management system providing process logic for key management system initialization located on the server;

a key management system storage providing a secure data storage for the key management system;

an application using the key management system to manage an application key;

an interface providing a means for inputting data into the key management system; and

a client computer operatively connected to the server, wherein the client computer comprises a user interface to manage the key management system.

[c27] A method for retrieving a value secured in a key management system comprising:

receiving a request for the value secured in the key management system;

searching for a key corresponding to the value in a decoded key list; and

retrieving a tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

[c28] The method of claim 27, wherein the key management storage is located on a second server.

[c29] The method of claim 27, wherein the key management system interface comprises a graphical user interface.

[c30] A method for retrieving a value secured in a key management system comprising:

receiving a request for the value secured in the key management system;

retrieving a serialized file from a key management system storage;

de-serializing the serialized file producing a de-serialized file;

decoding an encoded key list in the de-serialized file to produce a decoded key list;

searching for a key corresponding to the value in the decoded key list;

inputting a key encryption key into the key management system;

hashing the key encryption key to produce a key encryption key hash;

comparing the key encryption key hash to a hashed key encryption key in the de-serialized file;

decrypting a secret token in the de-serialized file using the key encryption key if the key encryption key hash is equal to the hashed key encryption key in the de-serialized file to produce at least one tuple;

storing the at least one tuple in a data structure within the key management system; and

retrieving the tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

[c31] The method of claim 30, further comprising:

searching a local file system, if the key corresponding to the value is not in the decoded key list.

[c32] A method for changing an existing key encryption key, comprising:

entering the existing key encryption key;

entering a new key encryption key;

de-serializing a serialized file producing a de-serialized file;

hashing the existing key encryption key producing a hashed key encryption key;

comparing the hashed key encryption key to a key encryption key hash in the de-serialized file;

decrypting a secret token using the existing key encryption key if the hashed key encryption key equals the key encryption key hash producing a tuple;

encrypting the tuple using the new key encryption key producing a new secret token;

hashing the new key encryption key producing a new hashed key encryption key; and

serializing the new hashed key encryption key and the new secret token to produce a new serialized file.

[c33] An apparatus for retrieving a value secured in a key management system comprising:

means for receiving a request for the value secured in the key management system;

means for searching for a key corresponding to the value in a decoded key list; and

means for retrieving a tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

[c34] A apparatus for retrieving a value secured in a key management system comprising:

means for receiving a request for the value secured in the key management system;

means for retrieving a serialized file from a key management system storage;

means for de-serializing the serialized file producing a de-serialized file;

means for decoding an encoded key list in the de-serialized file to produce a decoded key list;

means for searching for a key corresponding to the value in the decoded key list;

means for inputting a key encryption key into the key management system;

means for hashing the key encryption key to produce a key encryption key hash;

means for comparing the key encryption key hash to a hashed key encryption key in the de-serialized file;

means for decrypting a secret token in the de-serialized file using the key encryption key if the key encryption key hash is equal to the hashed key encryption key in the de-serialized file to produce at least one tuple;

means for storing the at least one tuple in a data structure within the key management system; and

means for retrieving the tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

[c35] An apparatus for changing an existing key encryption key, comprising:

means for entering the existing key encryption key;

means for entering a new key encryption key;

means for de-serializing a serialized file producing a de-serialized file;

means for hashing the existing key encryption key producing a hashed key encryption key;

means for comparing the hashed key encryption key to a key encryption key hash in the de-serialized file;

means for decrypting a secret token using the existing key encryption key if the hashed key encryption key equals the key encryption key hash producing a tuple;

means for encrypting the tuple using the new key encryption key producing a new secret token;

means for hashing the new key encryption key producing a new hashed key encryption key; and

means for serializing the new hashed key encryption key and the new secret token to produce a new serialized file.